ZAI-OD 建模指南

文档版本: 1.1 最后修改时间: 2019-2 作者: qq600585

目录

OD 的概念	2
OD 的技术指标	2
OD 建模步骤	3
准备环境	3
准备需要导入的图片	4
多显示器的优势	5
开始导入我需要训练的图片	6
开始框住我们需要检测的目标	9
使用多显示器操作检查框体正确性	9
检验框体	10
自动化调整框体尺度	11
用对齐工具自动生成框体	12
使用人脸对齐工具来生成人头的自动框体	13
使用.svm_od_marshal 工具制作框体	15
使用.svm_dnn_od 工具制作框体	15
使用.svm_od 工具制作框体	15
使用 LocalTrainingServer 训练 OD	16
测试 OD	22

OD 的概念

对象检测器在 ZAI 中简称 OD。当 OD 经过了高度训练后,它可以检测任何物体

OD 的技术指标

主要指标 运行平台:任何平台 并行:支持 运行时内存消耗:中小 运行时性能:低 CPU 需求:高 GPU 需求:无 超参数复杂度:傻瓜化 训练后的文件尺寸:极小

次要指标

残差网络支持:不支持 光照变换训练:不支持 深度学习支持:不支持 训练内存消耗:大规模训练内存消耗会巨大无比,小规模训练它会很快完成 训练时长:小规模很快,大规模很久 能支持的最大数据集训练:我最大测试了 5000 张图片, cpu 训练实在太慢了

OD 建模步骤

准备环境

需要配置 Al.conf,使用 Notepad++打开 Al.conf 首先确保 Key 是正确的 确保这里的 exe 和 dll 文件和 Al.conf 都在同一个目录下



确保 Key 的方法可以打开同一目录下的 LocalTrainingServer.exe

它会提示 key 的过期时间, key 所支持的技术体系,确保 OD_Key=True



准备需要导入的图片

提示:如果图片是原封不动的导入,一旦多了,内存消耗就会变得巨大无比,假如图片有 2000 张,即使 64G 内存也会告急。

技巧: PS 是最强大的图片处理工具,它不光是编辑强大,批处理也能非常简单暴力的对你要导入图片做减肥,合理的减肥不光能减少训练时间,也能节省出更多的内存和磁盘空间 关于 PS 的批处理技巧,请参考我撰写的 PS 自动脚本批处理。

打开 Z_AI_Model.exe

首次打开时,会提示 Key 的状态

如果看 accept Object Detector key, 就表示 OD 可以使用, 假如是 Reject Object Detector key, OD 将不会工作,发生这种情况,首先你需要确保 ai.conf 中的 key 正确的,接下来,检测你 的网络连接,比如 ping 一下 zpascal.net(key 服务器),看看网络畅通,然后重开 Z_AI_Model.exe



Prepare AI Engine 后面是使用的 AI 引擎,这里提示正在使用支持 cuda 加速的 AI 引擎 如果你的电脑不支持 cuda,你可以通过修改 ai.conf 将 ai 引擎指定使用 zAI_x64.dll



多显示器的优势

Z_AI_Model 中的 Log window 是必不可少的状态监视器,我们的训练,保存,打开,都会在 log window 中反应出来,假如使用多显示器,可以将 log window 拖到另一台显示器去监控



开始导入我需要训练的图片

2-Ai Moder Builder.							
Return IMGMatrix	Tools Menu			Ī	III Move	e 💿 Select	detecto
resolution: 1608 * 1028 fps	New	Ctrl+N					
-renderer optimized: 0:0:00	Open	Ctrl+O					_
	Save	Ctrl+S					
	Save as						
	Import	١٠	Import picture				
	export	►	Import .imgDause	et 👘			
	Alignment toolki	t 🕨	Import .AI_Set				
	Training Toolkit	•					
	Remote Files	F11					
	Label Editor	F12					
	Options						
	About	F1					
	Exit						

在 Tools Menu 中,我们通过 Import Picture 菜单来导入图片

在本文中,我选择导入一批人物图片

Model 对批量图片的导入是并行化的,当我们导入数千张图片时,cpu 会吃满,系统会以最高效率将图片从导入到 Model 工具来,Model 内存消耗也巨大

Open	Binary	-	-	-	-	-	-	-	-	-	▼ 4 ∳ Se	arch Binary
Organize 👻 New folder		_		_	_		_		_			11 · 🗆 🛛
Favorites Oestop Oestop Oestop Zoromitads Z345Downloads Destop	bear_sp.jpg	bear_sp_datas et9e0390f0cb 1a72d1db428 18a166499a	bear_sp_datas et708597b147 ff67e882ed47 728026b507	bear_sp_datas etacfe48c7f3d 14b08162c8f3 183ce9b31	bear.sp_datas etaf4327ba8d 47e4b4889131 cffefa0bcf.b	bear_sp_datas etbal8e14cfb 90244e69ade2 72b1fa4aa8	canglaoshi.b mp	canglaoshi.p	dog_dataset3 b8f9bb609b9 27d75b81d21 8fdefadfb.b	dog_dataset3 db3271f6df49 9d20a144606f e98bdf9.bmp	dog_dataset5 c27137f7dc2c cfe5388bfcc3 c3867b9.bmp	
Carl Ubranics Consents Music Consents Con	dog_dataset6 debc580028a 0de4e12dc3b 9dad0f65d	dog_dataset8 1323420c3fc7 66c9e4382508 63cff11.bmp	dog_datasetc c7b02fa35200 7a16a4c8573c 86d31bc.bmp	dog_datasetc d0c7b8307f4c 9a3e78a5ecb7 200aaf4.bmp	face (1).jpg	face (2).jpg	face (3).jpg	face (4).jpg	face (5).jpg	face (6).jpg	face (7).jpg	
Administrator Computer Computer Mac-MINI PASSBYOU-PC Computer Com	face (8).jpg	lena.bmp	person_face (1).jpg	person_face (2).jpg	person_face (3).jpg	person_face (4).jpg	person_face (5).jpg	person_face (6).jpg	person_face (7).jpg	person_face (8).jpg	person_face (9).jpg	
Generate Bin Git Project Git Project M Mov folder Section 215	person_face (10).jpg	person_face (11).jpg	person_face (12).jpg	person_face (13).jpg	person_face (14).jpg	person_face (15).jpg	person_face (16).jpg	person_face (17).jpg	person_face (18).jpg	person_face (19).jpg	person_face (20).jpg	
a secondo-iguvous a sec	person_face (21).jpg	person_face (22).jpg	person_face (23).jpg	person_face (24).jpg	surf_1.bmp	surf_2.bmp					re la	
File name "surf_2.bmp" 'face (1).jpg'	'face (2).jpg" "face (;	1).jpg" "face (4).jp	og" "face (5).jpg"	"face (6).jpg" "fac	e (7).jpg" "face (8).jpg" "lena.bmp"	"person_face (1)	jpg" "person_fac	e (2).jpg" "person	face (3).jpg" "pe	son_fac ▼ all p	icture (".bmp;"-yv12;".seq;" • Open Cancel

导入进来以后,系统将会自动做一次图片框体的排序



我们通过右方篮框小窗的提示可以看到,我们当前只有35张图片



在 Model 工具窗口的左上角,有两行绿色提示文字,分别代表

分辨率: 1687 * 1028 显卡渲染器帧率: 64(dx 锁帧 60) 流水线每秒发给 gpu 的指令数量:5064 Renderer optimized:0:24:0:0,

当前数据集的框体总数:渲染器剪裁(剪裁照片):单张照片最大框体:单张照片最小的框体





例 8:0:5:0 表示一共 8 个框体, 裁剪 0 个, 单张图片多的框体是 5 个, 最少的框体是 0 个

我们先删除刚才的框体,回到正题



开始框住我们需要检测的目标

先点 detector(检测器用的框体) 然后框照片里面的目标,我框的是个女孩的头像



由于框体操作是手动来的,很容易发生误操作,我们应该养成一个习惯:

使用多显示器操作检查框体正确性

每次当我们框完目标,都点一次 Label Editor,这里会罗列我们所有的框体,当我们看见没 有内容的东西就 X 干掉它



检验框体



现在我框了2个目标,一个是女孩头,一个是张朝阳

我们在 Label Editor 可以看出他们的框体比例还是差距很明显的 一个是接近方形,一个是长方形,这种差异,我将它称为框体尺度

框体尺度是个很重要的概念, 在训练时, 如果两个框体尺度不同, 训练 器就会发生迭代错误, 并且不会提示, 这很可能会造成我们的训练出来 的模型不符合预期。zAI 通过调整框体尺度工具来解决该问题



自动化调整框体尺度

使用菜单中的 Calibration Scale Space 矫正 1:1,表示宽:高都是相同比例 1:1,宽=1,高=1 0.5:1,表示宽只占高的 1/2,宽=0.5,高=1 1:0.5,表示宽比例是高的 2 倍,宽=1,高=0.5



我们使用 1:1 矫正框体,完成后,我们在 Label Editor 可以看出两个框体尺度已经相同了

制作数据集的过程是反反复复的,框体校正会步步为营



用对齐工具自动生成框体

我们先通过 Model 窗口的 auto 按钮鸟瞰全数据集



这里面的人头真不少,我们一个一个去框,实在是太累了



使用人脸对齐工具来生成人头的自动框体

这里有两个 face 对齐工具,他们的作用是生成对齐人脸图+框体+人脸对齐几何 这两个工具一个是放大以后对齐,这可以生成很小的人脸对齐数据,只要图片中包含了完整 的人脸,它在 99%的情况下都会生成正确数据。

一个是快速对齐,使用同比例尺度对齐,不过多余的人脸对齐预处理,这种预处理出来的框体大多是比较清晰的框体,质量更好



Face Alignment 是并行化的 cpu 计算,在运行期间,电脑的 cpu 会全部满负荷,耐心等待一会,它就会将人脸全部检测出来了



我们再次使用 Label Editor 来查看检测出来的人头

这是我们需要的结果吗?肯定会有很多框体质量不好,如图像模糊不清,这时候,我们需要将不合格的图像干掉



点 X 即可





使用.svm_od_marshal 工具制作框体

.svm_od_marshal 是 OD 集合,会多出标签,需要一个已训练完成的数据文件。 这是高级内容,本文作为入门指南,对此不做过多介绍

使用.svm_dnn_od 工具制作框体

.svm_dnn_od 工具需要 GPU 支持,,会多出标签,需要一个已训练完成的数据文件。 这是高级内容,本文作为入门指南,对此不做过多介绍

使用.svm_od 工具制作框体

.svm_od 工具需要一个已训练完成的数据文件。

这是高级内容,本文作为入门指南,对此不做过多介绍

使用 LocalTrainingServer 训练 OD

概念:将 OD 交给一台服务器去训练。

Model 工具链会自动将我们的数据打包,发给 LocalTrainingServer 训练,待 LocalTrainingServer 训练完成后,我们再从 LocalTrainingServer 将训练结果取出来使用。训练结果,是一个后缀.od_svm 的文件。

首先,确保 LocalTrainingServer 已经启动,注意看 OD_Key:True 的提示



在 Model 工具中,通过 Options 来设置 LocalTrainingServer 的 IP 地址, IP 地址可以是 IPV6, 也可以是公网



当我们确定了 LocalTrainingServer 正确的 IP 地址以后 我们打开训练工具参数



只要勾上了 Send Training source to Local Server.工具链就会自动使用远程服务器进行训练 训练模型的参数叫做超参数,这些参数,可以通过研究 zAI 的 OD 训练 Demo 来了解



开始 Run Training

这时候,Log window 会开始滚动显示训练进度,这些训练都在 LocalTrainingServer 服务器干的,我们看到的都是服务器的状态

Log window	۲
objective: 2.89875 objective gap: 1.99988 risk: 2.01535 risk gap: 1.99988 num planes: 12 iter: 10	*
objective: 2.29154 objective gap: 1.37768 risk: 1.39367 risk gap: 1.37768 num planes: 13 iter: 11	
objective: 2.80649 objective gap: 1.73994 risk: 1.75553	Ţ

因为本文的 OD 是很小的数据集,训练很快,大概 1 分钟以后,会看到一条提示 训练任务已经完成了,是否要立即从 LocalTrainingServer 将训练结果下载下来并且打开



我们点 Yes, 然后, Log Window 上会有个进度提示, 等它走完



然后,工具链会弹出一个文件窗口,这里面就是训练的结果



下面,来一一讲解这些文件集的用途

Package: OD-document_task.output												
New	Open	Save	Save as	Save as .OXC								
Root			Name									
			param.txt input.img[output.sv result.txt log.txt engine_log LastOutpu	: Dataset m_od g.txt ut.txt								

- Param.txt,这是我们刚才编辑给训练用的超参数
- Input.imgDataset,这是我们的发给服务器的数据集,这个数据集是可以用 z_ai_model.exe 直接打开编辑
- Output.svm_od,这是训练的结果,.svm_od 训练好的数据,它可以直接挂接在 zAI 引擎 上做目标对象识别
- Result.txt,这是训练结果报告,里面是诸如耗了多少时间,是否训练成功
- Log.txt,这是训练过程的 log 记录
- Engine_log.txt,这是训练过程的 zAI 引擎 log 记录
- lastOutput.txt,这是输出的文件列表

后缀是.txt 的,我们通过双击,会直接用记事本打开查看

如果我们要使用 output.svm_od,我们就将它导出成文件

同样的,如果我们要使用 input.imgDataset,我们也将它导出来,然后用 z_ai_model.exe 打 开编辑



🐵 Packag	e: OD-docu	ment_tasl	c.output							- 🗆	×
New	Open	Save	Save as	Save as .OXC	MD5	0b4d9434	450f3f330278b2d5fe	ecf818e	File Size:20. IO Write:0	24M IO Read: 70. 15Kb	
Root			Name			Property	Size		Time	Modification	Time
			param.txt			0000000	122	2019/	2/21 22:01:27	2019/2/21 2	2:01:27
			input.img[ataset		00000000	20.10M	2019/	2/21 22:01:27	2019/2/21 2	2:01:27
			output.sv	n_od		0000000	68.15Kb	2019/	2/21 22:01:27	2019/2/21 2	2:01:27
			result.txt			00000000	59	2019	2/21 22:01:27	2019/2/21 2	2:01:27
			log.txt			00000000	49.73Kb	2019/	2/21 22:01:27	2019/2/21 2	2:01:27
			engine_log	,.txt		00000000	25.03Kb	2019/	2/21 22:01:27	2019/2/21 2	2:01:27
		LastOutput.txt				00000000	63	2019/	2/21 22:01:27	2019/2/21 2	2:01:27
										*	
open DB:C:\ export file:c: export file:c: export file:C	CoreLibrary\ :\temp\result :\temp\LastC :\Users\Adm	zAI\Binary\C t.txt Dutput.txt inistrator\De)D-document :sktop\docum	_task.output ent_task.svm_od							

每当我们查看.txt 或则导出 output.svm_od,都可以在下方看到提示信息

测试 OD

新创建一个 model,



重新把图片都导入进来,不需要存盘





在对齐工具使用 OD market,载入刚才训练的 od_svm 即可

对齐工具会以并行方式工作,5秒以后,这批图片就被重新框了,其实这是我们的预期结果。 OD测试也就完成了。



完

By.qq600585 2018-9